IN THE CLAIMS:

Claim 1 is amended herein. Claim 16 is added. All pending claims and their present status are produced below.

1. (Currently Amended) A computer based system for switching between program contexts comprising:

an embedded pipelined processor capable of having a first program thread and a second program thread in an execution pipeline;

a first set of data storage devices capable of storing a first state of said embedded processor;

a second set of data storage devices capable of storing a second state of said embedded processor; and

a thread scheduler for identifying which of said program threads said embedded processor executes and configurable to allocate available processing time of the embedded pipelined processor among at least the first and second states according to a fixed schedule;

wherein said processor switches between said first and second state ~~in a time period between~~ after the end of the execution of a first program instruction in the first thread and before the beginning of the execution of a second program instruction ~~in the second thread~~.

2. (Original) The system of claim 1, wherein said first state is the state of the embedded processor during the execution of the first program thread.

3. (Original) The system of claim 1, wherein said second state is the state of the embedded processor during the execution of the second program thread.

4. (Original) The system of claim 1, wherein said processor switches between said first and second states by changing a state selection register.

5. (Original) The system of claim 1, wherein said thread scheduler includes:

a thread identifier for identifying at least one hard-real-time (HRT) thread and at least one non-real-time thread;

a HRT scheduler for regularly scheduling said HRT thread in available time quanta such that said HRT thread is scheduled to ensure the execution of the HRT in a predetermined time.

6. (Original) The system of claim 5, wherein said time quanta is at least one instruction cycle.

7. (Original) The system of claim 5, wherein said thread scheduler schedules a non-real-time (NRT) thread to replace a scheduled HRT thread if said HRT is complete.

8. (Original) The system of claim 5, wherein said thread scheduler schedules the execution of non-real-time (NRT) threads in quanta not allocated to HRT threads.

9. (Original) The system of claim 8, wherein said thread scheduler regularly schedules NRT threads to be executed.

10. (Original) The system of claim 5, further comprising:

a first storage device for storing program instructions, said processor fetching instructions from the first storage device within a first fetch period;

a second storage device for storing program instructions, said processor fetching instructions from the second storage device within a second fetch period;

B2 6 wherein said first fetch period is substantially shorter than said second fetch period.

1 11. (Previously Added) The system of claim 10, wherein said first storage device for
2 storing program instructions comprises a static RAM.

1 12. (Previously Added) The system of claim 10, wherein said second storage device
2 for storing program instructions comprises a flash memory.

1 13. (Previously Added) The system of claim 1, wherein said embedded processor is
2 capable of restoring said second state of said embedded processor during execution of said
3 first program thread.

1 14. (Previously Added) The system of claim 1, wherein said embedded processor is
2 capable of storing said second state of said embedded processor during execution of said first
3 program thread.

1 15. (Previously Added) The system of claim 1, wherein said first set of data storage
2 devices comprises registers shared by a plurality of threads.

1 16. (New) The system of claim 1, wherein the fixed schedule is one of a fixed strict
2 schedule, a semi-flexible strict schedule, and a loose strict schedule.